

**REMARKS**

In view of the foregoing amendments and following remarks responsive to the Office Action dated August 11, 2004, Applicant respectfully requests favorable reconsideration of this application.

The Office has rejected all claims under 35 USC section 102 (e) as being anticipated by Anders. Applicant respectfully traverses.

**The Present Invention**

The present invention is a method and apparatus for loading web pages, including supplemental files such as pictures, sound files, video files, etc., at a browser. One of the problems of the prior art addressed by the present invention is that browsers typically read the HTML code in a Web page from left to right and from top to bottom. Accordingly, the browser encounters the embedded references to such supplemental files in the order in which they are encountered while reading the page. Naturally, the browser typically will send requests back to the server for those supplemental files in the order it encounters the references while reading the HTML code. Since a browser has a limited number of ports, the supplemental files may not be retrieved and loaded in the most efficient manner. For instance, if a browser has four ports and the requested page has 14 supplemental files, in which the first four referenced supplemental files are large files and the next 10 are small files, the browser may take a long time to download the first four large files, while the person sitting at the

client browser watches a blank screen. If the ten small files could be downloaded first, the browsing experience for the person can be much improved because he/she could then have something to look at while waiting for the four large files to download.

The present invention addresses this concern without the need to modify the browser software in any way. In accordance with the invention, the order in which supplemental files referenced in a Web page are downloaded from the server to the requesting client is specified by the designer of the HTML code of the Web page and controlled at the server side regardless of the order in which the client-side Web browser encounters and requests the supplemental files. Particularly, each supplemental file referenced in a Web page has a sequence number associated with it. In a preferred embodiment, the sequence number is provided as an additional attribute of the tag that calls the supplemental file. Since the client Web browser is a standard Web browser, it will have no idea what the sequence attribute is, but that does not matter because a browser will simply ignore any attribute within a tag that it does not understand. However, at the server-side, when the page is requested, the server parses the page before sending it to the requesting client to find the tags for the supplemental files embedded within the page and reads the associated sequence number attributes. It then builds a queue for serving the supplemental files to the client machine, the supplemental files being queued in the order dictated by the sequence numbers.

Thereafter, regardless of the order in which the browser returns requests for the supplemental files, the server will serve the supplemental files in the order dictated by the queue. Existing browsers already are equipped to receive and cache files and associate such cached files with files referenced in an HTML page. Accordingly, the fact that the supplemental files referenced in a Web page may be received in an order different from the order in which the browser requests them is of no consequence. Accordingly, the invention resides entirely at the server side and will work with any of the widely available Web browsers.

#### **The Anders Reference**

Anders discloses a method and apparatus for serving Web pages, including supplemental files, to a requesting client. However, the method and apparatus disclosed in Anders is entirely different from that of the present invention. Most notably, unlike the present invention, Anders requires the Web browser software to be modified to function with the invention. See col. 8, lines 2-6 (indicating the need for a Jammer unpacker “on the client”), col. 8, lines 51-54, and col. 12, line 65 – col. 14, line 7 (which describes in detail the software needed at the browser to implement the invention).

Anders’ scheme is entirely different than Applicant’s. With reference to Anders’ Figure 8, the server transmits the requested page to the requesting client in a particular data stream format 190 that includes the data for the main object (the Web page) and

the data for the supplemental objects (such as embedded pictures, etc.) in data entries (packets) that are interleaved with each other in an order selected by the developer. More particularly, the data stream 190 comprises a stream header 180 at the beginning of the stream followed by data definition entries and HTML data entries. Each data definition entry, e.g., 181, 182, 185, 187, defines a supplemental object present in the Web page data stream. There is one data definition entry per file. The HTML data entries are simply the actual data of the files (including the main file as well as the supplemental files). Each file will typically consist of many HTML data entries that the browser assembles together to render the whole file. The data definition entry that defines any given file must precede the first HTML data entry of that file in order for the browser to know what to do with those HTML data entries when it receives them.

The basic premise of Anders' invention is that the browser, upon receiving each data definition entry, creates an entry in an unpacked object cache (UOC). Then, when the browser starts receiving the HTML data entries corresponding to the supplemental file identified by any given data definition entry, it will append that HTML data to the entry it created in its UOC. In Anders, the browser receives the tags identifying the supplemental files in the order dictated by the data stream 190. Accordingly, the browser may be receiving data of a supplemental file before it receives the HTML data entry that contains the reference to that supplemental file. That is not a problem. Particularly, when the browser reaches the reference to the supplemental file that it has

already started downloading and caching in its UOC, the UOC simply forwards the cached data to the browser for rendering.

### **Discussion**

While Anders discloses an interesting technique, it is entirely different from the present invention. In Anders, there is nothing that resembles the sequence number attribute embedded within the tag referencing the supplemental file. Furthermore, the server does not parse the code being sent to the client to detect the sequence numbers. There are no sequence numbers. Rather, Anders' server builds the data stream 190 using a software module that Anders calls the publisher 210 (see figure 11).

While Anders' technology does permit the server to dictate the order in which supplemental files are delivered to the browser, it does so in a way that is entirely different from what is claimed in the present application.

Referring to claim 1, Anders does not disclose (1) "parsing the code comprising the requested page to detect data within the code that indicates an order in which said supplemental files are to be served" or (2) "constructing a queue indicating said order".

Claims 2-8 depend from claim 1 and, therefore, distinguish over Anders for at least all of the reasons given above in connection with claim 1.

However, in addition, the dependent claims add even further distinguishing features. For instance, claim 6 depends from claim 1 and adds that the references to the supplemental files "comprise HTML tags, and said order data comprises attributes

of said tags". There is nothing in Anders remotely resembling this. Claim 8 depends from claim 7 and further adds that "said order data attributes are not recognizable by said client machine". This is directly contrary to Anders, in which the client machine must be modified in accordance with Anders' technology in order to recognize Anders' data stream.

Independent claim 9 also distinguishes over Anders. Claim 9 includes the limitation of "second code indicating an order in which said supplemental files are to be rendered, said second code associated with each of said references and comprising an attribute of a tag associated with said supplemental file". Hence, claim 9 distinguishes over Anders for at least all of the reasons discussed above in connection with claims 1 and 6. Dependent claim 11 further distinguishes over Anders by further describing the nature of the HTML tag comprising the sequence number. Anders, which does not have a sequence number at all, obviously cannot teach such limitations.

Independent claim 12 also distinguishes over Anders by virtue of reciting "program code for parsing said code defining the Web page to detect said order data", and "program code for constructing a queue in a memory, said queue comprising a list of said supplemental files in said order".

Claims 12-18 depend from claim 11 and, therefore, distinguish over Anders for at least all of the same reasons as independent claim 11.

In addition, dependent claim 17 adds "said references to supplemental files comprise HTML tags" and that "said order data comprises attributes of said tags".

As discussed above in connection with claim 6, these limitations are not found in Anders. Claim 18 depends from claim 17 and further adds that "said order data attributes are not recognizable by said client machine". This is not found in Anders as discussed above in connection with claim 7.

Independent claim 19 includes the limitations "code for parsing said code defining a Web page to detect said order data", "code for constructing a queue in a memory, said queue comprising a list of said supplemental files in said order", and "code for serving said supplemental files to said requesting client machine in said order of said queue". Accordingly, claim 19 also patentably distinguishes over Anders for essentially all of the same reasons discussed above in connection with the other independent claims.

Claims 19-22 depend from claim 18 and, therefore, distinguished over Anders for at least all of the reasons set forth above in connection with claim 18.

In addition, dependent claims 21 and 22 recite essentially the same subject matter as previously discussed in connection with dependent claims 6 and 7, respectively. Accordingly, they even further distinguish over the prior art for the same reasons given above in connection with dependent claims 6 and 7.

In view of the foregoing amendments and remarks, this application is now in condition for allowance. Applicant respectfully requests the Examiner to issue a Notice of Allowance at the earliest possible date. The Examiner is invited to contact

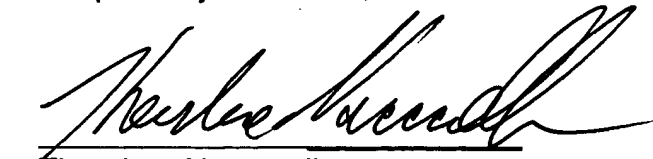
**APPLICATION NO. 09/823,331**

**DOCKET NO. P-24,723 USA**

Applicant's undersigned counsel by telephone call in order to further the prosecution of this case in any way.

Respectfully submitted,

Dated: 11.12.04

  
Theodore Naccarella  
Registration No. 33,023

Synnestvedt & Lechner LLP  
2600 Aramark Tower  
1101 Market Street  
Philadelphia, PA 19107  
Telephone: (215) 923-4466  
Facsimile: (215) 923-2189

Attorney for Applicant